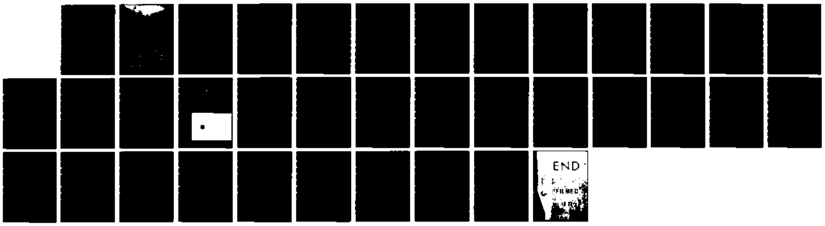
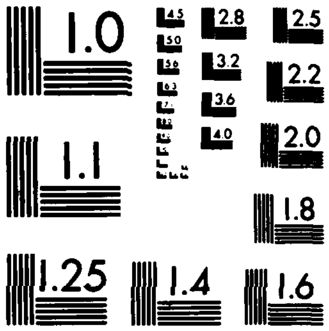


END
FILMED


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

(10)

ADA132375

rochester

Department of Computer Science
University of Rochester
Rochester, New York 14627

83 09 13 017

12

Hough Transformation into Cache Accumulators: Considerations and Simulations

Christopher M. Brown and David B. Sher
Computer Science Department
University of Rochester
Rochester, NY 14627

TR114
August 1982

Abstract

The Hough transform provides a paradigm for multi-dimensional pattern detection and more general parameter extraction. It has good properties in noise, but as usually implemented requires an accumulator array whose size is exponential in the number of parameters extracted. The idea of accumulating votes in a small content-addressable store raises many technical issues, some of which are outlined here. Simulation results illustrate some of the points.

Index terms: Hough transform, shape detection, pattern recognition, Cache implementation.

The preparation of this paper was supported in part by the Office of Naval Research under Grants N00014-80-C-0197 and N00014-82-K-0193, and in part by the Defense Advanced Research Projects Agency under Grant N00014-78-C-0164.

1. Introduction

The *Hough Transform* (HT) maps a point of one parameter space A (often a local feature space or generalized image) into a set of compatible points of another space B (often a space of "higher-level," more global parameters) [Hough 1962; Duda and Hart 1972; Ballard 1981]. This process is sometimes called *voting* (the feature points vote for possible higher-level constructs of which they could be a part).

HT is traditionally implemented by accumulating votes in a quantized version of the parameter space B, called an *accumulator array*. A naive implementation of the accumulator array uses space exponential in the number of dimensions (parameters)--this makes multi-parameter HT impractical. One approach to this problem is to quantize the parameter space dynamically [Sloan 1981; O'Rourke 1981]. Another idea, the subject of this report, is the *cache-based HT*, which uses a fixed (small) content addressable store (in software, a hash table; in hardware, a cache) to accumulate votes.

To study the properties of the cache-based HT, it is helpful to have an abstract, domain-free characterization of the HT vote-generation process. Existing applications (e.g., [Brown and Curtiss 1982]) and theory (e.g., [Shapiro 1975, 1978a, 1978b; Brown 1982]) suggest considerations that should be taken into account by an abstraction. The abstraction may be analyzed or simulated with various caching schemes to derive a model of cache-based HT performance. Then, existing applications (such as our shape, motion, and illuminant-direction computations) may be used to test model predictions.

Some considerations for a general model of the cache-based HT process appear in Sections 2 and 3, and Section 4 gives an abstraction of the voting process. Section 5 outlines some possible analytical approaches. Section 6 presents results of pilot studies on cache behavior. Figure 1 is a diagram of the system for analysis of the cache-based HT.

<<Figure 1>>

2. Informal Abstract Hough Transform

This section lists considerations that govern the behavior of any HT like process emitting votes through time into an accumulator. Such an abstract process is called a *voting machine* in Section 4.

In a noiseless HT, one feature point in A produces a set H of votes that generally are weighted points in the parameter space B (so a vote is a vector (w, b), with w a weight and b in B. (We assume here that the HT is "tuned" to one parameter transformation, to detect instances of one shape, etc.) In one limiting case, the set H is unbounded (as when a point on a line votes for an infinite line of points in (slope,intercept) space). In another limiting case, H is a singleton (as in the work of Shapiro [1975, 1978a, 1978b]). In either case, for noiseless HT we assume that H from a feature point in A contains exactly one vote for the "true" parameters in B that characterize the instance in A. Let us call the true parameter value the *peak* or mode for the instance, since that is where the plurality (mode) of votes for it should be. H is the *feature point spread function* (fpsf) of [Brown 1982]. In all that follows, we assume accumulation into a quantized version of B, so H = fpsf is a discrete set. Of course H may be a complicated function of the parameter values and locations in the originating parameter space A: H = fpsf = fpsf(x, A(x)). As A is scanned, individual feature points cause votes to be emitted. The summation of all H from an instance into the accumulator array is the *parameter point spread function* (ppsf) (Figure 2).

<<Figure 2>>

In the work reported here, the fpsfs (Hs, voting patterns) are of two types: 1) all votes have uniform unit weight; 2) each vote (of weight 2) is associated with an "inhibitory surround"--two votes of weight -1. We call the former case the *regular* HT. The latter scheme is a variant of the *C'Hough* technique [Brown 1982, Brown and Curtiss 1982].

(Consideration 2.1) Instances in an image consist of several feature points, extend across several scan lines, and are spatially coherent. There may be more than one feature on a scan line. Instances may overlap.

(Consideration 2.2) The image can be scanned in any of several ways: e.g., left right, top-bottom; pixels at random (with and without replacement); left right across scan lines in random order; left-right across interlaced scan lines.

(Consideration 2.3) Votes come in bursts of length ||fpsf(x, A(x))|| = ||H||, each generated by a single feature vector in A.

(Fact) With one instance and no noise, successive bursts all come from the single instance. With multiple instances and no noise, successive bursts may or may not come from the same instance. The likelihood that they do depends on assumptions about the distribution of instances in the image (do they overlap?) about the distribution of feature points in an instance (is there more than one per scan line?), and about the scanning pattern. (The basic question is whether two features of the same instance are encountered before another instance.)

(Consideration 2.4) Only one (weighted) vote in H is a vote for the true instance parameters (for the peak or mode).

(Consideration 2.5) A basic question in HT is its performance in noise. H's from noise points are interleaved with H's from instances. There are several noise phenomena with differing characteristics and implications. First there is *statistical noise.* One form is signal-independent image degradation. This process establishes a kind of baseline, since its random nature makes it one of the least pernicious forms of noise in the cache-based HT. One component of degradation, perhaps worth singling out, is that component consisting of features not associated with any instance in A. This component adds extraneous H's, but does not affect H's from true instances.

Another form of noise is errors in computing "image features," [illegible] the feature vectors in A. These errors, as well as *location* and *quantization noise* in A, may be modeled as certain random processes.

The most pernicious form of noise, and one not treated here, is *systematically* misleading instance information. For instance, extreme and systematic feature dropout can arise from occlusion. Especially troublesome are near miss instances (many features in common with instances of interest). The problem is that in the cache-based HT, it is important to create and retain peaks for the instances. These peaks are threatened by competition from other growing peaks, such as arise in near miss instances.

3. Informal Abstract Cache

A hash table or cache accepts one element at a time, is content addressable, and garbage collection, or *flushing*, happens when its finite length is exhausted. Unlike a cache for memory management, the HT cache is not meant to track a dynamic situation, but to function as a "smart accumulator."

The Cache has a fixed length. If that is infinite, the cache acts like the accumulator array. The Cache is often most easily addressed as a linear data structure, with vote vectors transformed to offset integers. Addressing modes may vary in hardware implementations.

An intuitively appealing cache-flushing strategy is to flush entries with fewest votes. There are many variations, including using recency information (traditional for LRU caches) and geometric locality information (implemented through a hierarchical cache arrangement). For applications, the flush algorithm may reflect considerations of hardware implementation. The treatment of negative votes in CHough raises questions for the flush algorithm.

Our cache is a data structure and associated operations for management of a set S from the space of weighted vectors from B. Thus elt = (w, b) for some w $\in$ Reals (or w $\in$ Integers) and **b** $\in$ B. The *length* of a cache is the number of elements it can hold. As do most abstract data types dealing with sets, a cache has basic initialization and housekeeping operations, as well as operations of the following flavor (these are for exposition, not practicality).

Full(cache): TRUE if cache full, else FALSE.
FindVector(searchelt,cache, @foundelt): sets foundelt to unique element whose vector agrees with that of searchelt and returns TRUE, or returns FALSE if no such element is in cache.
FindVote(weight, cache, @foundeltSet): sets foundeltSet to set of elements whose weight agrees with (or is less than or equal to) weight and returns TRUE, or returns FALSE if no such element is in cache.
FindMax(cache, @foundelt): sets foundelt to the element (or one of the elements) with maximum weight.
Remove(elt, cache): removes elt from cache.
Insert(elt, cache): adds elt (not in cache) to non-Full cache.
AddWeight(elt, cache): does nothing but return FALSE if no element matching elt's vector is in cache. Otherwise it increments the matching vector's weight

by the weight of elt and returns TRUE.

There are three high-level cache-management operations: Vote, Flush, and Interpret.

Vote(elt, cache): Vote uses AddWeight to increment the weight (vote count) of an existing element, or else Inserts element elt into the cache if there is room, or as a last resort invokes a more or less complicated method of dealing with a full cache, usually involving Flush and possibly re trying the Insert. One feature point causes ||H|| votes.

Flush(cache): Flush creates room for more elements. Its design is a focus of current research (Section 6). Algorithms with hardware implementation are especially of interest.

Interpret(cache): Interpret is to identify "peaks" in accumulator space B, which it is hoped correspond to (are the parameters of) instances in the input space A. If the input contains only a single instance, the natural Interpret returns the element of maximum weight (most votes). For multiple instances, especially an unknown number of them, Interpret presents interesting problems, and is a focus of current research. As with Flush, algorithms with a natural hardware implementation are especially of interest.

4. A General Voting Machine

A geometric imaging-like situation is an easy way to describe the voting machine, and in fact the most straightforward implementation probably is as shape detection applied to an image-like array. A description as some form of stochastic finite state machine would work equally well (and may be easier to analyze), but isn't as visualizable.

The *abstract image* is a two-dimensional array A(x) of vector entries corresponding to instances and to noise. The voting machine scans this array, and for each feature vector it encounters it emits an H = fpsf(x,A(x)). Houghing for a global parameter (like sun angle) is modeled by an A(x) with only one instance. Houghing for multiple instances is possible. Noise (Consideration 2.5) may be modeled.

The following technical decisions arose from geometric encodings of the considerations in Section 2 and from [Brown 1982].

1) Let an instance be one or more vertical lines (called *parts*) in a horizontal row, separated by empty columns. This captures all facets of Consideration 2.1.

2) Various forms of scanning (Consideration 2.2) are implemented by accessing A in different orders.

3) The fpsf's for an instance should look like diameters of a circle (Figure 2). The HT of an instance (the contents of the accumulator array) will then be a radial rosette of lines all passing through a central point, the peak. This decision is based on Considerations 2.3 and 2.4, and the fact that this choice reproduces the ubiquitous near-peak 1/r falloff of parameter psfs in all dimensions [Brown 1982]. For C Hough,

the line of positive votes should have an "inhibitory surround" of negative votes associated with it in the fpsf.

4) Noise can take various forms (Consideration 2.5). Missing instance points can, if systematic, lead to non radially symmetric ppsfs. Noise feature vectors can be added into the A(x), and of course instance feature vectors can be perturbed by noise processes. In the experiments reported here, we had no massive dropout (modeling occlusion), getting a similar effect by varying the number of parts per instance. We did not attempt to address questions of near-miss instances at this stage. For this report we used the baseline case of signal-independent image degradation.

The variables that characterize the abstract voting machine are:

V1) voting pattern for each feature (fpsf)
V2) number of instances and their location
V3) number of parts/instance
V4) noise parameters
V5) scanning method

Figure 3 shows an A(x) with two 2-part instances with and without noise. Each part is 9 features long; there are eighteen different features.

<<Figure 3>>

5. Analytic Approaches

1) Hypothesis Testing: After the HT, the accumulator holds a distribution of votes. After accumulating the votes in a cache, the cache is meant to mirror certain characteristics of the accumulator's distribution. In particular, it is meant to have the same modes (local maxima). One could proceed with something like a statistical hypothesis testing approach, with the null hypothesis being that modes in the cache correspond to modes in the accumulator. The idea of course is to measure the probabality of Type I and Type II errors (falsely rejecting and falsely accepting the null hypothesis), given some sample statistic--in our case, the cache contents. Usually, we choose probabilities α and β for these errors that are small enough for our purposes. When things go well (i.e., in textbooks) this approach yields a critical (rejection) region of values of the test statistic for which we should reject the null hypothesis if we are to maintain our standards for error probabilities. There are non-parametric tests for things like the sameness of median between two distributions.
What goes wrong immediately here is that the sampling performed by the cache is extremely structured. Even if we were to attempt to compute with the known relevant distributions [Brown 1982], the intricacies of this sampling seem very resistant to analysis.
A generalization of the hypothesis testing approach is a decision theory approach, which allows a more flexible cost function to be associated with decisions, and which can incorporate empirical data. Sequential decision theory seems possibly more relevant, though less well understood.

2) Page Replacement, etc.: There is a long tradition of work (e.g., the analysis of

page replacement) aimed at characterizing cache behavior. Almost certainly none of that work is directly relevant, but it may provide useful analytical tools and abstractions.

3) Geometry and Probability: Last (this seems most promising), the particular abstraction chosen to model cache-based HT might be analyzed with very straightforward probability and statistical tools. The model proposed in Section 4 is very geometrical in flavor, corresponding to throwing line segments sequentially down on the plane and analyzing their accumulating statistics of overlap. One major hurdle is incorporating the effects of the flush algorithm into the accumulation of statistics.

6. Results

6.1 Overview

We implemented the abstract voting machine of Section 4 and a software simulation of a cache that is parameterizable in several ways, including its length and flush algorithm. The cache is instrumented for various forms of statistical analysis. The results are presented in tables and figures that follow this section. They are preceded first by a high level overview of the experiments, then a more detailed explanation of the parameters involved.

We ran a set of experiments to explore several parameters (both of the cache and of the abstract image) governing sequential HT performance.

Fixed Parameters:	Flush Algorithm (Threshold)
	Number of Instances (1)
	Part Length (11)
Varying Parameters:	Cache length (32, 64, 128, ∞)
	Number of Parts (1, 2, 3)
	Noise (0, 15%, 30%)
	Fpsf Length (3, 9, 15 cells)
	Fpsf (Regular Hough, CHough)
Scanning method	(five methods)

The results of these experiments are summarized in Tables 1-8, which show a "peak/maximum background ratio" measure and its variation for each case. The ratio is a form of signal-to-noise ratio, and is computed as follows.

$$\text{Ratio} = \frac{\text{weight of vector describing instance}}{\text{maximum weight of all other vectors}} \qquad (1)$$

<<Tables 1-8>>

Thus a ratio of 3.00 means that the single parameter vector correctly describing the instance is three times as high as its nearest competitor. A ratio of unity means there is a tie. Any ratio greater than unity means the correct peak would be found by simply choosing the highest peak. A ratio less than unity means such simple thresholding finds a vector giving an incorrect instance description.

We expanded a subset of these results in more detail.

Fixed Parameters: Flush Algorithm (as above)
Number of Instances (as above)
Part Length (as above)
Number of parts (1)

Varying Parameters: Cache length (as above)
Noise (15%, 30%)
Fpsf Length (as above)
Scanning method (two methods)

The expanded results are shown in Figure 4, which gives the histograms of the peak/background ratio whose means and standard deviations appear in Tables 1-4. From the histograms it can be seen, for instance, what fraction of the time an instance would not be detected by a simple thresholding version of the Interpret function.

<<Figure 4>>

A smaller number of the CHough results of Tables 5-8 (those with cache lengths of 64 and infinity) were expanded into histograms (Figure 5).

<<Figure 5>>

Finally, Figure 6 shows histograms like those of Figures 4 and 5. Here the main focus is on the flush algorithm as cache length and noise level vary.

<<Figure 6>>

6.2 Details

The choice of which parameters to vary and interesting values for them was determined by a significant amount of prior experiment. The size of the abstract image is irrelevant, being redundant with other parameters such as noise density. In fact the image was 20 x 20, and the single instance was located in its center.

Flush Algorithms: We used Threshold Flush exclusively in the main experiments. In regular Hough, it simply raises a threshold for the vote count (weight) of elements until it finds elements whose weight is at threshold, all of which it flushes. With CHough, it similarly raises the threshold from zero until elements are found, but then all elements of that weight and below (including all those with negative vote weights) are flushed. For the last experiment, we compared Regular Hough under Threshold Flush with Regular Hough under Random Below Threshold Flush, which simply finds an element at random whose weight is in the bottom nth percentile of weights and Removes that single element. In this experiment the doomed element was picked at random from the bottom third of weighted elements.

Number of Instances: Only one instance appears in images for these experiments. This is an important case, and fairly terse statistics describe the efficacy of the simple-threshold Interpret algorithm. Multiple instance performance is a matter for future research (also see [Brown and Curtiss 1982]).

Number of Parts: One-, two-, and three-part instances generate (in the noiseless case) that number of feature points per horizontal scanline.

Part Length: Part length 11 means the instance extends across 11 horizontal scan lines. The total "signal strength" for the instances is 11, 22, and 33 votes.

Cache length (32, 64, 128, infinite): The infinite cache is a pure accumulator array. This variable can to some extent be traded off against Fpsf Length. It is worth mentioning that commercially available hardware has cache lengths on the order of thousands of words.

Noise (0%, 15%, 30%): These experiments use a noise model that mimics signal independent image degradation. A random feature vector overwrites an abstract image point (instance or background) with the indicated probability.

Feature point spread function (fpsf) Length (3, 9, 15 cells): The fpsf for Regular Hough is a line of 1's, produced by a simple DDA algorithm. The corresponding fpsf for CHough is a similar line of 2's, flanked on either side by a line of -1's. The resulting ppsfs for 13-feature instances are shown in Figure 2b and c. Short fpsfs produce less inherent noise (lower sidelobes) [Brown 1982]. For example, consider line detection: As better and better edge gradient information is incorporated to reduce the possible range of slope attributed to the line, the fpsf shrinks. Inherent noise is not a problem in single-instance images, but the effects of the statistical noise also felt at greater distance with larger fpsfs, and that accounts for the performance degradations noticable here.

Scanning method: We use five methods to scan the abstract image. The first four hit each image point exactly once, the last is not guaranteed to. The first two are deterministic, the last three are random. The first, fourth, and fifth might be easily implemented in hardware.

1) Left to Right within Top to Bottom: as in a non interlaced TV scan.
2) Top to Bottom within Left to Right: interesting because it causes several fpsfs from the instance to arrive almost successively.
3) Random without replacement: implemented by accessing the n^2 elements of the nxn array in permuted order.
4) Left to Right within randomly permuted scanlines.
5) Random with replacement: Access n^2 elements of the nxn image array at random, very likely missing some entirely and consequently hitting others more than once.

N: There are two sources of randomness: One arises from noise, one from the random scanning methods. In the regular Hough experiment we ran ten different noisy images into each scanning method. The random methods each produced ten different orders of scan, the scanline methods each produced one. Thus in the finite cache regular Hough experiments, the N for the two non-random scanning methods (1 and 2) is 10, and for the three random scanning methods (3, 4, 5) is 100. In the CHough experiment, we simply generated 20 images and submitted them to the scanning algorithms, getting (presumably) different random scans for each. In the infinite cache, the four methods (1,2,3,4) that hit each element once are equivalent, so are collapsed into one case.

6.3 Conclusions and Future Work

How should these results be interpreted? To apply them to any particular HT situation (but so far only to detection of a single instance or d[illegible]ction of a parameter vector for a single phenomenon, and only with signal inde[illegible]dent noise), analyze it in terms of scanning algorithm, number of feature points, n[illegible]level, fpsf length and so forth. Interpolate or extrapolate as necessary. Reme[illegible]r that the important basic process is the succession of votes emerging from the [illegible]'s mix of feature and noise elements). The explicit conversion from any given i[illegible]ation to parameters of our model will be addressed in more detail in a future report (or a later edition of this report).

The infinite cache (a pure accumulator array) is not dramatically better than the finite caches under the circumstances of these experiments. Thus finite caches seem to be practical HT accumulators.

The tables and histograms show a graceful and qualitatively predictable degradation of performance as noise and fpsf length increase and cache length decreases.

In good conditions (short fpsf, low noise, large instances), average CHough performance is reliably markedly better than regular Hough. However, as conditions get worse, its average performance degenerates to approximately that of the regular HT. The variation in the peak maximum background ratio appears markedly higher for the CHough schemes, but this is mostly explained by the doubling of the peak height in this CHough implementation (Figure 2b, c).

The random (with replacement) scan was uniformly significantly worse in all cases than other methods. The random (without replacement) does not seem significantly better. In fact, the results are (perhaps surprisingly) insensitive to scanning method, as long as each input space point is hit once.

Even in 30% noise, the Top to Bottom within Left to Right scanning method was not better than the Left to Right within Top to Bottom. This indicates that successive bursts of votes (fpsfs) containing votes for the "true parameter" are not enough to help the peak establish itself. Probably the noise that arrives before the instance and the flush algorithm dominate the peak-establishment process.

Several of the histograms show a significant bimodality with one peak at 0. This indicates that the "true peak" never gets established in a significant number of cases.

and that some pre-processing might help dramatically in establishing peaks. One idea is "pair Houghing," in which only vectors that appear in two lpsts are allowed into the cache. Clearly without such preprocessing the issue is whether peaks are able to survive in the cache, and this depends to a large extent on the Flush algorithm. Instances found early in the scan might have an advantage in some flushing strategies, while other strategies might penalize peaks that have been in the cache longer. We reasoned that the simple Threshold Flush might be systematically flushing nascent peaks before they got established, thus accounting for the significant number of 0 peak/maximum background ratios.

Jerry Feldman suggested the Random Below Threshold Flush to replace the systematic "slaughter of innocents" imposed by Threshold Flushing. The result is a lottery-like system that selects from a range of vote weights, allowing some small peaks to survive. As shown in Figure 5 (for regular Hough only), this method makes for a significant improvement in mean peak/maximum background ratio and reduces the frequency of occasions on which the correct vector ends up with zero vote weight.

Systematic distortion, near-miss instances, and occlusion pose thorny problems for the cache-based HT. We emphasize that many of the conclusions of this report and all the quantitative data are based on a statistical, signal independent noise model. An important topic for future work is to assess the performance of cache HT schemes in these more stringent noise conditions and to develop means of improving such performance.

Hierarchical caches can incorporate a form of "geometric locality" into flushing. Natural flush algorithms for a single-level cache are based strictly on weights. Multiple caches in a resolution pyramid, maintained in parallel, can be used to restrict flushing to coherent localities in the input space A. This idea is another matter for future research.

References

Ballard, D.H., "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition 13*, 2, 111-122, 1981.

Brown, C.M., "Bias and noise in the Hough transform I: Theory," TR 105, Computer Science Dept., U. Rochester, June 1982; submitted to *IEEE Trans. on Pattern Analysis and Machine Intelligence.*

Brown, C.M. and M. Curtiss, "Bias and noise in the Hough transform II: Experiments," TR113, Computer Science Dept., U. Rochester, August 1982.

Duda, R.O. and P.E. Hart, "Use of the Hough transform to detect lines and curves in pictures," *CACM 15*, 1, 11-15, 1972.

Hough, P.V.C., *Method and Means for Recognizing Complex Patterns*, U.S. Patent 3069654, 1962.

O'Rourke, J., "Dynamically quantized spaces for focusing the Hough transform," *Proc.*, 7th IJCAI, 737-739, Vancouver, B.C., August 1981.

Shapiro, S.D., "Feature space transforms for curve detection," *Pattern Recognition 10*, 129-143, 1978a.

Shapiro, S.D., "Generalization of the Hough Transform for curve detection in noisy pictures," *Proc.*, 4th Int'l. Joint Conf. on Pattern Recognition, 710-714, Kyoto, Japan, November 1978b.

Shapiro, S.D., "Transformations for the computer detection of curves in noisy pictures," *Computer Graphics and Image Processing 4*, 328-338, 1975.

Sloan, K.R., Jr., "Dynamically quantized pyramids," *Proc.*, 7th IJCAI, Vancouver, B.C., August 1981.

Table and Figure Captions

Tables 1, 2, 3, 4, 5, 6, 7, 8: Mean and standard deviation of peak/maximum background ratio (Eq. 1) for regular (1, 2, 3, 4) and CHough (5, 6, 7, 8) schemes. Table rows are scanning methods: l) Left to Right within Top to Bottom. u) Top to Bottom within Left to Right. p) Random without replacement. r) Random with replacement. lp) Left to Right within permuted scanlines. For explanation of other parameters, see text.

Figure 1: The cache-based Hough transform analysis system.

Figure 2: Several parameter point spread functions (ppsfs): the accumulation of all weighted votes in space B arising from a parameter instance in space A. (a) The results of [Brown 1982] suggest this general form for a ppsf in the abstract voting machine. Each fpsf is a straight line segment, and together they form a radial rosette with their intersection forming the peak. This configuration exhibits many of the properties of general n-dimensional ppsfs. (b) The design of (a) implemented for the abstract voting machine. This is the regular Hough ppsf for a 13-feature instance. The fpsfs are of length 9, each a single segment of unit weight votes. (c) The CHough version of (b). The fpsfs are a segment of 9 votes of weight 2, flanked by segments of weight -1. Note the reduced sidelobes (off peak element weights). To be directly comparable with (b), vote weights should be divided by two. (d) Ppsfs for regular HT (left) and CHough (right) arising in the application of circle location. The displays show the absolute value of sidelobe height coded as intensity (the peaks are the same height). Near the peak these ppsfs are well approximated by the abstract voting machine ppsfs.

Figure 3(a-b): (a) Two 18-feature, two-part instances in a 20x20 abstract image. (b) The instances of (a) with 15% signal-independent noise.

Figure 4(a-p): Histograms of peak/maximum background ratios from a subset of regular HT cases from Tables 1-4. Random scan is random with replacement.

Figure 5(a-f): As in Figure 4, for CHough (Tables 5-8). Random scan is random with replacement.

Figure 6(a-d): Comparison of two Flush algorithms. Histograms as in Figures 4 and 5.

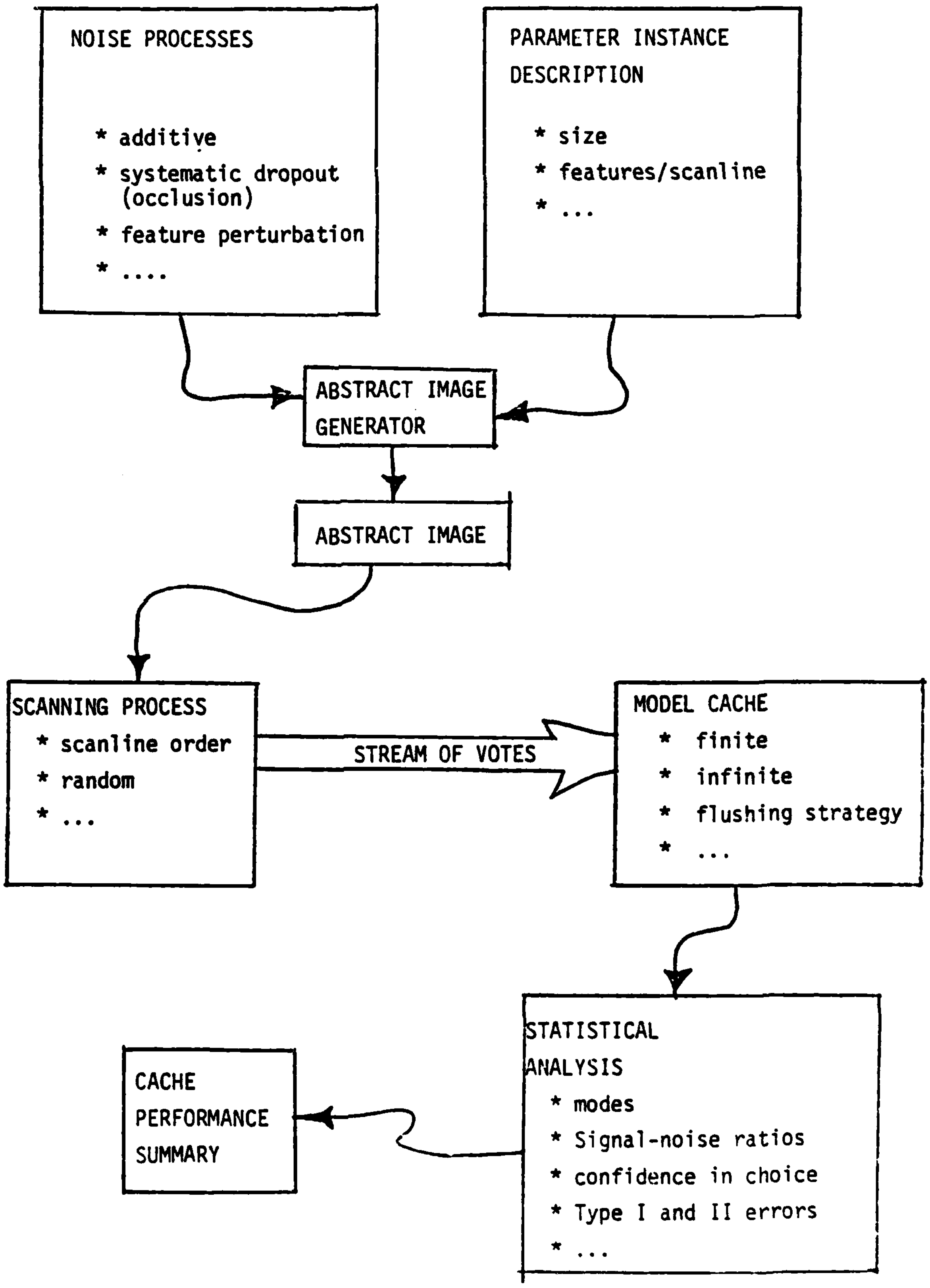
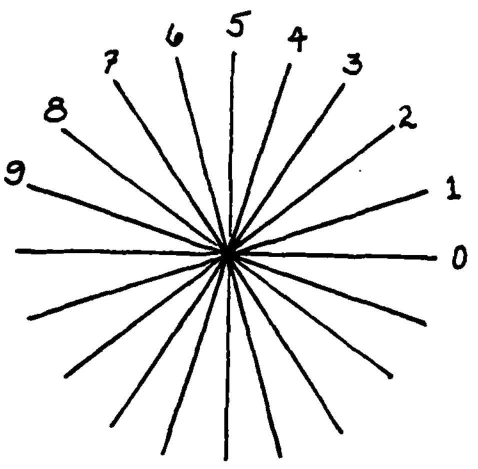

NOISE PROCESSES
* additive
* systematic dropout (occlusion)
* feature perturbation
* ....
PARAMETER INSTANCE DESCRIPTION
* size
* features/scanline
* ...
ABSTRACT IMAGE GENERATOR
ABSTRACT IMAGE
SCANNING PROCESS
* scanline order
* random
* ...
STREAM OF VOTES
MODEL CACHE
* finite
* infinite
* flushing strategy
* ...
STATISTICAL ANALYSIS
* modes
* Signal-noise ratios
* confidence in choice
* Type I and II errors
* ...
CACHE PERFORMANCE SUMMARY

Figure 1.

(a)



(b)

.	.	.	.	.	.	.	.	.	.	.
.	1	1	1	.	2	.	1	1	1	.
.	.	1	1	1	2	1	1	1	.	.
.	1	1	1	2	2	2	1	1	1	.
.	1	1	1	3	4	3	1	1	1	.
.	1	1	3	3	13	3	3	1	1	.
.	1	1	1	3	4	3	1	1	1	.
.	1	1	1	2	2	2	1	1	1	.
.	.	1	1	1	2	1	1	1	.	.
.	1	1	1	.	2	.	1	1	1	.
.	.	.	.	.	.	.	.	.	.	.

(c)

.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	-1	-1	.	.	.	-1	-1	.	.	.
.	.	2	1	1	-4	4	-4	1	1	2	.	.
.	-1	-1	1	.	-1	.	-1	.	1	1	-1	.
.	.	.	.	1	1	-4	1	1	.	.	.	.
.	-1	.	.	-2	-4	5	-4	-2	.	.	-1	.
.	.	.	-2	.	2	26	2	.	-2	.	.	.
.	-1	.	.	-2	-4	5	-4	-2	.	.	-1	.
.	.	.	.	1	1	-4	1	1	.	.	.	.
.	-1	-1	1	.	-1	.	-1	.	1	-1	-1	.
.	.	2	1	1	-4	4	-4	1	1	2	.	.
.	.	.	-1	-1	.	.	.	-1	-1	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.

(d)

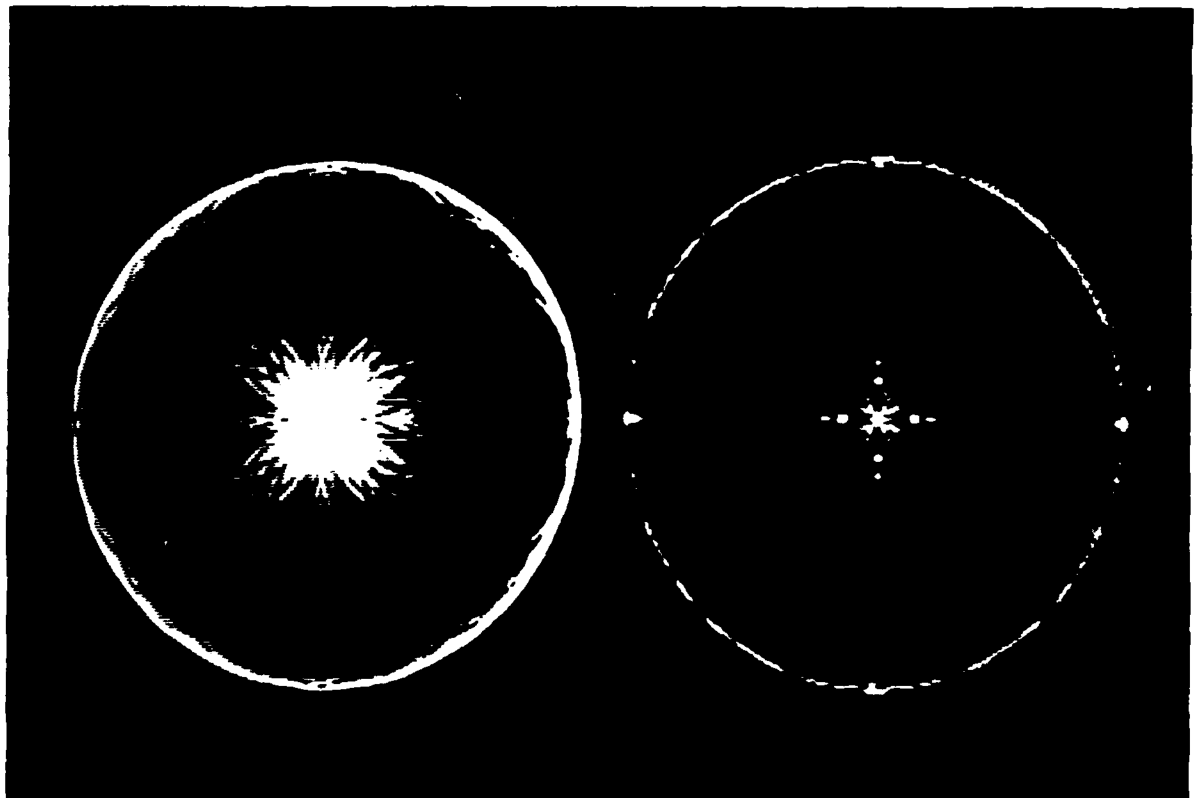
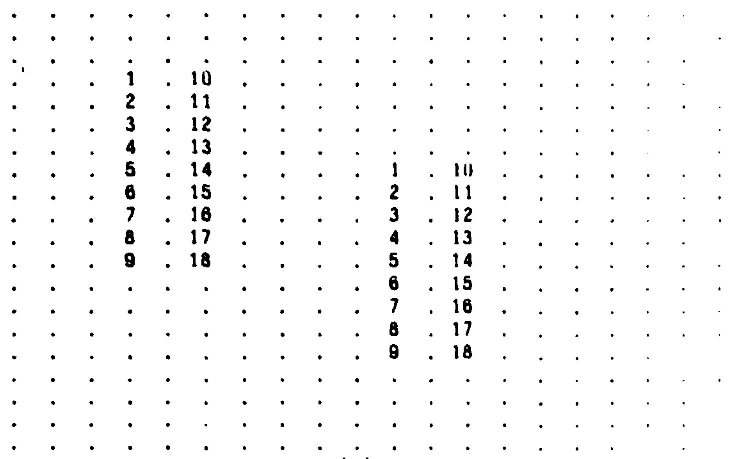


Figure 2.


(a)

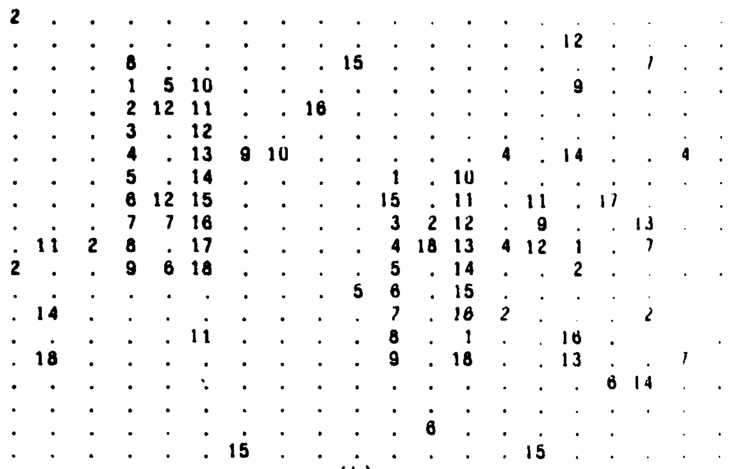

(b)

Figure 3.

Cache length: 32. Part Length: 11

Number of parts: 1

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg:l)	2.75	2.75	2.75	2.49	1.15	0.26	1.29	0.06	0.27
StDev:	0.00	0.00	0.00	0.78	0.84	0.46	0.85	0.13	0.53
N:	10	10	10	10	10	10	10	10	10
Avg:u)	2.75	2.75	2.75	2.31	1.84	1.42	1.48	0.56	0.00
StDev:	0.00	0.00	0.00	0.46	0.28	0.61	0.54	0.64	0.00
N:	10	10	10	10	10	10	10	10	10
Avg:p)	2.75	2.99	4.05	2.71	1.58	0.88	1.76	0.41	0.64
StDev:	0.00	0.46	2.01	0.85	0.88	0.97	0.85	0.60	0.68
N:	100	100	100	100	100	100	100	100	100
Avg:r)	2.37	2.30	2.61	1.67	0.98	0.57	1.01	0.30	0.47
StDev:	0.61	0.60	0.74	0.76	0.72	0.66	0.78	0.50	0.58
N:	100	100	100	100	100	100	100	100	100
Avg:lp)	2.75	3.06	3.68	2.67	0.94	0.42	1.42	0.19	0.37
StDev:	0.00	0.49	1.43	0.85	0.93	0.80	0.98	0.48	0.63
N:	100	100	100	100	100	100	100	100	100

Number of parts: 2

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg:l)	3.14	3.14	3.14	2.96	3.23	2.76	2.74	2.17	1.51
StDev:	0.00	0.00	0.00	0.56	0.78	1.42	0.92	0.93	1.29
N:	10	10	10	10	10	10	10	10	10
Avg:u)	3.14	3.14	3.14	2.71	2.46	2.60	2.35	1.23	0.77
StDev:	0.00	0.00	0.00	0.29	0.30	0.64	0.42	0.97	0.96
N:	10	10	10	10	10	10	10	10	10
Avg:p)	3.14	3.23	3.52	3.13	2.83	2.35	2.85	1.74	1.16
StDev:	0.00	0.22	0.72	0.84	0.78	0.92	1.01	0.91	0.98
N:	100	100	100	100	100	100	100	100	100
Avg:r)	2.55	2.77	2.91	2.46	2.09	1.56	1.87	1.22	0.74
StDev:	0.40	0.57	0.69	0.55	0.67	0.88	0.70	0.83	0.77
N:	100	100	100	100	100	100	100	100	100
Avg:lp)	3.14	3.60	3.53	3.38	3.25	2.76	2.97	2.21	1.45
StDev:	0.00	0.66	0.70	1.17	0.87	1.09	1.22	0.76	1.12
N:	100	100	100	100	100	100	100	100	100

Number of parts: 3

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg:l)	3.30	3.30	3.30	2.98	3.49	3.33	2.97	2.84	2.55
StDev:	0.00	0.00	0.00	0.22	0.74	1.30	0.38	0.51	1.12
N:	10	10	10	10	10	10	10	10	10
Avg:u)	3.30	3.67	3.67	2.92	3.01	2.86	2.83	2.33	0.91
StDev:	0.00	0.00	0.00	0.18	0.46	0.47	0.33	1.11	1.22
N:	10	10	10	10	10	10	10	10	10
Avg:p)	3.30	3.53	3.82	3.14	3.05	2.88	3.11	2.64	2.04
StDev:	0.00	0.41	0.62	0.39	0.63	0.80	0.81	0.78	0.99
N:	100	100	100	100	100	100	100	100	100
Avg:r)	2.93	2.95	3.19	2.65	2.47	2.21	2.40	2.03	1.38
StDev:	0.40	0.45	0.63	0.53	0.73	0.92	0.58	0.79	1.04
N:	100	100	100	100	100	100	100	100	100
Avg:lp)	3.30	3.48	8.79	3.22	3.54	3.51	3.35	3.02	2.25
StDev:	0.00	0.37	8.69	0.58	0.87	1.03	0.99	0.68	1.08
N:	100	100	100	100	100	100	100	100	100

TABLE 1

Cache length: 64. Part length: 11.

Number of parts: 1

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg:l)	2.75	2.75	2.75	2.38	1.52	1.29	1.86	0.35	0.50
StDev:	0.00	0.00	0.00	0.50	0.65	0.77	0.46	0.56	0.56
N:	10	10	10	10	10	10	10	10	10
Avg:u)	2.75	2.75	2.75	2.30	1.82	1.61	1.75	1.01	0.69
StDev:	0.00	0.00	0.00	0.48	0.32	0.35	0.26	0.33	0.51
N:	10	10	10	10	10	10	10	10	10
Avg:p)	2.75	2.75	3.03	2.47	1.76	1.41	1.88	0.85	0.89
StDev:	0.00	0.00	0.42	0.56	0.45	0.65	0.52	0.52	0.62
N:	100	100	100	100	100	100	100	100	100
Avg:r)	2.37	2.22	2.35	1.58	1.20	0.90	1.11	0.55	0.65
StDev:	0.61	0.47	0.52	0.51	0.48	0.58	0.61	0.51	0.53
N:	100	100	100	100	100	100	100	100	100
Avg:lp)	2.75	2.75	3.01	2.44	1.67	1.08	1.87	0.55	0.72
StDev:	0.00	0.00	0.41	0.55	0.65	0.84	0.55	0.66	0.69
N:	100	100	100	100	100	100	100	100	100

Number of parts: 2

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg:l)	3.14	3.14	3.14	2.75	2.53	2.39	2.21	2.19	1.88
StDev:	0.00	0.00	0.00	0.30	0.32	0.46	0.32	0.48	0.46
N:	10	10	10	10	10	10	10	10	10
Avg:u)	3.14	3.14	3.14	2.70	2.53	2.51	2.38	1.96	1.67
StDev:	0.00	0.00	0.00	0.25	0.22	0.40	0.42	0.31	0.64
N:	10	10	10	10	10	10	10	10	10
Avg:p)	3.14	3.14	3.19	2.81	2.65	2.40	2.47	2.03	1.64
StDev:	0.00	0.00	0.17	0.35	0.46	0.58	0.59	0.47	0.57
N:	100	100	100	100	100	100	100	100	100
Avg:r)	2.55	2.63	2.66	2.31	2.15	1.85	1.82	1.43	1.17
StDev:	0.40	0.42	0.49	0.43	0.50	0.51	0.52	0.56	0.59
N:	100	100	100	100	100	100	100	100	100
Avg:lp)	3.14	3.14	3.22	2.80	2.82	2.56	2.39	2.15	1.74
StDev:	0.00	0.00	0.21	0.36	0.55	0.68	0.48	0.45	0.55
N:	100	100	100	100	100	100	100	100	100

Number of parts: 3

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg:l)	3.30	3.30	3.30	2.94	2.75	2.98	2.86	2.66	2.35
StDev:	0.00	0.00	0.00	0.18	0.20	0.46	0.31	0.37	0.48
N:	10	10	10	10	10	10	10	10	10
Avg:u)	3.30	3.30	3.30	2.91	2.68	2.87	2.74	2.43	1.12
StDev:	0.00	0.00	0.00	0.18	0.30	0.47	0.28	0.17	1.15
N:	10	10	10	10	10	10	10	10	10
Avg:p)	3.30	3.31	3.42	2.95	2.77	2.71	2.86	2.67	2.27
StDev:	0.00	0.05	0.18	0.21	0.34	0.42	0.34	0.36	0.49
N:	100	100	100	100	100	100	100	100	100
Avg:r)	2.93	2.84	2.98	2.54	2.27	2.28	2.29	2.10	1.77
StDev:	0.40	0.37	0.44	0.43	0.46	0.54	0.47	0.48	0.65
N:	100	100	100	100	100	100	100	100	100
Avg:lp)	3.30	3.30	3.46	2.95	2.91	2.97	2.94	2.80	2.35
StDev:	0.00	0.00	0.29	0.23	0.49	0.62	0.42	0.46	0.39
N:	[illegible]	100	100	100	100	100	100	100	100

TABLE 2

Cache length: 128. Part length: 11.

Number of parts: 1

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg:l)	2.75	2.75	2.75	2.30	1.69	1.55	1.73	1.15	0.84
StDev:	0.00	0.00	0.00	0.48	0.22	0.38	0.29	0.22	0.66
N:	10	10	10	10	10	10	10	10	10
Avg:u)	2.75	2.75	2.75	2.30	1.74	1.64	1.73	1.13	1.12
StDev:	0.00	0.00	0.00	0.48	0.23	0.32	0.29	0.26	0.33
N:	10	10	10	10	10	10	10	10	10
Avg:p)	2.75	2.75	2.75	2.30	1.75	1.58	1.78	1.03	1.16
StDev:	0.00	0.00	0.00	0.48	0.30	0.37	0.33	0.39	0.37
N:	100	100	100	100	100	100	100	100	100
Avg:r)	2.37	2.22	2.33	1.57	1.25	1.05	1.14	0.76	0.84
StDev:	0.61	0.47	0.49	0.47	0.38	0.44	0.49	0.39	0.38
N:	100	100	100	100	100	100	100	100	100
Avg:lp)	2.75	2.75	2.75	2.31	1.76	1.55	1.77	1.00	1.12
StDev:	0.00	0.00	0.00	0.48	0.37	0.44	0.33	0.46	0.50
N:	100	100	100	100	100	100	100	100	100

Number of parts: 2

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg:l)	3.14	3.14	3.14	2.71	2.54	2.18	2.18	1.93	1.74
StDev:	0.00	0.00	0.00	0.29	0.29	0.33	0.28	0.25	0.38
N:	10	10	10	10	10	10	10	10	10
Avg:u)	3.14	3.14	3.14	2.71	2.52	2.31	2.19	1.95	1.75
StDev:	0.00	0.00	0.00	0.29	0.25	0.38	0.28	0.31	0.29
N:	10	10	10	10	10	10	10	10	10
Avg:p)	3.14	3.14	3.14	2.72	2.53	2.28	2.22	2.00	1.72
StDev:	0.00	0.00	0.00	0.29	0.33	0.39	0.33	0.32	0.36
N:	100	100	100	100	100	100	100	100	100
Avg:r)	2.55	2.63	2.58	2.28	2.01	1.80	1.76	1.45	1.28
StDev:	0.40	0.42	0.42	0.41	0.38	0.42	0.43	0.48	0.39
N:	100	100	100	100	100	100	100	100	100
Avg:lp)	3.14	3.14	3.14	2.71	2.56	2.32	2.23	2.01	1.76
StDev:	0.00	0.00	0.00	0.29	0.36	0.42	0.32	0.35	0.34
N:	100	100	100	100	100	100	100	100	100

Number of parts: 3

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg:l)	3.30	3.30	3.30	2.92	2.60	2.56	2.75	2.50	2.28
StDev:	0.00	0.00	0.00	0.18	0.16	0.31	0.27	0.26	0.40
N:	10	10	10	10	10	10	10	10	10
Avg:u)	3.30	3.30	3.30	2.92	2.63	2.59	2.75	2.47	2.28
StDev:	0.00	0.00	0.00	0.18	0.19	0.33	0.27	0.22	0.31
N:	10	10	10	10	10	10	10	10	10
Avg:p)	3.30	3.30	3.31	2.92	2.61	2.57	2.78	2.52	2.20
StDev:	0.00	0.00	0.05	0.19	0.22	0.33	0.29	0.26	0.35
N:	100	100	100	100	100	100	100	100	100
Avg:r)	2.93	2.84	2.89	2.53	2.17	2.13	2.23	1.98	1.77
StDev:	0.40	0.37	0.37	0.43	0.39	0.40	0.43	0.40	0.42
N:	100	100	100	100	100	100	100	100	100
Avg:lp)	3.30	3.30	3.30	2.92	2.60	2.60	2.77	2.53	2.24
StDev:	0.00	0.00	0.00	0.18	0.21	0.35	0.28	0.27	0.36
N:	100	100	100	100	100	100	100	100	100

TABLE 3

Cache Length: Infinite. Part Length: 11.

Number of parts: 1

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg: l, u,p,lp)	2.75	2.75	2.75	2.30	1.67	1.59	1.73	1.13	1.33
StDev:	0.00	0.00	0.00	0.48	0.24	0.31	0.29	0.17	0.21
N:	10	10	10	10	10	10	10	10	10
Avg:r)	2.37	2.22	2.33	1.57	1.25	1.09	1.14	0.83	0.92
StDev:	0.61	0.47	0.49	0.47	0.34	0.33	0.49	0.30	0.25
N:	10	10	10	10	10	10	10	10	10

Number of parts: 2

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg: l, u,p,lp)	3.14	3.14	3.14	2.71	2.43	2.16	2.18	1.89	1.69
StDev:	0.00	0.00	0.00	0.29	0.25	0.30	0.28	0.26	0.28
N:	10	10	10	10	10	10	10	10	10
Avg:r)	2.55	2.63	2.58	2.28	1.96	1.74	1.75	1.43	1.27
StDev:	0.40	0.42	0.42	0.41	0.34	0.37	0.42	0.43	0.31
N:	10	10	10	10	10	10	10	10	10

Number of parts: 3

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg: l, u,p,lp)	3.30	3.30	3.30	2.92	2.57	2.46	2.75	2.44	2.12
StDev:	0.00	0.00	0.00	0.18	0.18	0.25	0.27	0.21	0.27
N:	10	10	10	10	10	10	10	10	10
Avg:r)	2.93	2.84	2.89	2.53	2.16	2.08	2.22	1.94	1.71
StDev:	0.40	0.37	0.37	0.43	0.39	0.37	0.43	0.38	0.38
N:	10	10	10	10	10	10	10	10	10

TABLE 4

```
Cache Length: 32.  Part Length: 11.  CHOUGH

Number of parts: 1

Noise %     0%                |   15%               |    30%              |
Fpsf Len:   3      9      15  |    3      9     15  |     3      9     15 |

Avg:l)    4.40   3.67   3.67  |  3.14   1.09   0.88 |  1.95   0.57   0.29 |
StDev:    0.00   0.00   0.00  |  0.93   1.04   0.96 |  0.89   0.76   0.65 |
N:          20     20     20  |    20     20     20 |    20     20     20 |

Avg:u)    4.40   3.67   3.67  |  2.88   2.09   1.62 |  2.03   1.40   0.92 |
StDev:    0.00   0.00   0.00  |  0.65   0.45   0.53 |  0.70   0.65   1.02 |
N:          20     20     20  |    20     20     20 |    20     20     20 |

Avg:p)    4.51   4.90   6.03  |  3.15   1.70   1.10 |  2.16   1.16   0.36 |
StDev:    0.67   1.29   2.80  |  0.79   1.14   0.89 |  1.22   0.92   0.67 |
N:          20     20     20  |    20     20     20 |    20     20     20 |

Avg:r)    4.23   3.20   3.33  |  2.21   0.97   0.37 |  1.04   0.21   0.58 |
StDev:    1.90   0.84   1.95  |  1.00   0.94   0.48 |  1.04   0.50   1.01 |
N:          20     20     20  |    20     20     20 |    20     20     20 |

Avg:lp)   4.57   4.89   5.79  |  3.19   0.95   0.61 |  1.67   0.49   0.41 |
StDev:    0.39   1.33   2.56  |  0.90   1.06   0.85 |  1.26   0.72   0.83 |
N:          20     20     20  |    20     20     20 |    20     20     20 |


Number of parts: 2

Noise %     0%                |   15%               |    30%              |
Fpsf Len:   3      9      15  |    3      9     15  |     3      9     15 |

Avg:l)    6.29  11.00  14.67  |  5.75   4.63   3.72 |  5.13   2.58   1.47 |
StDev:    0.00   0.00   0.00  |  1.97   1.02   1.64 |  1.45   1.00   1.14 |
N:          20     20     20  |    20     20     20 |    20     20     20 |

Avg:u)    6.29   4.40   4.40  |  3.95   4.18   3.27 |  3.41   2.30   1.91 |
StDev:    0.00   0.00   0.00  |  0.86   1.29   0.90 |  0.53   1.16   0.85 |
N:          20     20     20  |    20     20     20 |    20     20     20 |

Avg:p)    6.03   5.20   6.58  |  4.38   4.48   3.21 |  4.31   2.80   1.07 |
StDev:    0.47   0.88   1.53  |  1.36   1.41   2.21 |  1.54   0.95   0.94 |
N:          20     20     20  |    20     20     20 |    20     20     20 |

Avg:r)    4.40   3.46   4.48  |  2.52   2.46   2.40 |  2.80   1.37   0.77 |
StDev:    1.63   1.20   1.66  |  0.61   0.86   1.71 |  1.01   1.06   0.90 |
N:          20     20     20  |    20     20     20 |    20     20     20 |

Avg:lp)   6.25   6.49   7.60  |  5.25   4.74   3.94 |  4.54   2.82   1.53 |
StDev:    0.17   1.11   2.20  |  1.86   1.37   1.29 |  1.40   0.95   0.82 |
N:          20     20     20  |    20     20     20 |    20     20     20 |


Number of parts: 3

Noise %     0%                |   15%               |    30%              |
Fpsf Len:   3      9      15  |    3      9     15  |     3      9     15 |

Avg:l)    6.00  11.00  33.00  |  5.30   5.94   6.04 |  5.59   4.38   3.16 |
StDev:    0.00   0.00   0.00  |  1.50   1.98   1.61 |  1.47   0.97   1.53 |
N:          20     20     20  |    20     20     20 |    20     20     20 |

Avg:u)    6.00   5.08   4.40  |  4.34   4.24   4.80 |  4.01   3.46   2.92 |
StDev:    0.00   0.00   0.00  |  0.84   1.17   1.72 |  0.72   1.08   1.48 |
N:          20     20     20  |    20     20     20 |    20     20     20 |

Avg:p)    6.10   6.17   7.22  |  5.23   5.19   5.51 |  5.23   3.94   2.31 |
StDev:    0.31   0.76   2.07  |  1.78   1.49   2.17 |  1.93   1.38   1.50 |
N:          20     20     20  |    20     20     20 |    20     20     20 |

Avg:r)    5.26   5.18   5.22  |  3.94   3.32   3.51 |  3.00   2.64   2.01 |
StDev:    1.69   1.85   1.96  |  1.32   1.28   1.95 |  1.20   1.54   1.51 |
N:          20     20     20  |    20     20     20 |    20     20     20 |

Avg:lp)   5.97   7.15  18.02  |  5.53   6.09   5.80 |  5.39   4.27 [illegible] |
StDev:    0.11   2.10   4.13  |  1.78   1.92 [illegible] | [illegible] [illegible] 1.68 |
N:    [illegible] 20 [illegible] | [illegible] [illegible] [illegible] |
```

TABLE 5

```
Cache Length: 64.  Part Length: 11.  CHOUGH.

Number of parts: 1

Noise %    0%                |   15%                |   30%                |
Fpsf Len:  3      9      15  |    3      9      15  |    3      9      15  |

Avg:l)   4.40   4.40   3.67  |  3.02   1.87   1.16  |  2.00   0.74   0.68  |
StDev:   0.00   0.00   0.00  |  0.71   0.61   0.82  |  0.70   0.64   0.82  |
N:         20     20     20  |    20     20     20  |    20     20     20  |

Avg:u)   4.40   4.40   3.67  |  2.99   2.07   1.82  |  2.00   1.40   1.34  |
StDev:   0.00   0.00   0.00  |  0.62   0.53   0.43  |  0.63   0.43   0.45  |
N:         20     20     20  |    20     20     20  |    20     20     20  |

Avg:p)   4.51   4.41   4.54  |  3.19   2.03   1.43  |  2.05   1.04   0.77  |
StDev:   0.67   0.65   0.79  |  0.72   0.82   0.79  |  0.75   0.70   0.63  |
N:         20     20     20  |    20     20     20  |    20     20     20  |

Avg:r)   4.23   3.14   2.73  |  2.06   1.06   0.80  |  1.24   0.48   0.81  |
StDev:   1.90   0.97   0.75  |  0.74   0.83   0.54  |  0.85   0.44   0.77  |
N:         20     20     20  |    20     20     20  |    20     20     20  |

Avg:lp)  4.57   4.58   4.20  |  3.11   1.82   1.23  |  2.01   0.79   0.87  |
StDev:   0.39   0.94   1.00  |  0.80   0.53   0.72  |  0.78   0.63   0.69  |
N:         20     20     20  |    20     20     20  |    20     20     20  |


Number of parts: 2

Noise %    0%                |   15%                |   30%                |
Fpsf Len:  3      9      15  |    3      9      15  |    3      9      15  |

Avg:l)   6.29   6.29  11.00  |  4.88   4.10   3.71  |  4.21   2.60   1.86  |
StDev:   0.00   0.00   0.00  |  1.24   1.01   0.79  |  0.94   0.61   0.67  |
N:         20     20     20  |    20     20     20  |    20     20     20  |

Avg:u)   6.29   5.50   4.40  |  4.10   3.89   3.13  |  3.59   2.45   2.19  |
StDev:   0.00   0.00   0.00  |  1.31   0.84   0.46  |  0.77   0.45   0.59  |
N:         20     20     20  |    20     20     20  |    20     20     20  |

Avg:p)   6.03   5.51   5.51  |  4.32   3.91   3.52  |  4.02   2.68   1.97  |
StDev:   0.47   0.94   1.05  |  1.25   1.07   0.71  |  1.00   0.81   0.70  |
N:         20     20     20  |    20     20     20  |    20     20     20  |

Avg:r)   4.40   3.47   4.09  |  2.34   2.44   2.39  |  2.47   1.51   1.17  |
StDev:   1.63   1.01   1.03  |  0.51   0.76   1.08  |  0.86   0.62   0.62  |
N:         20     20     20  |    20     20     20  |    20     20     20  |

Avg:lp)  6.25   6.32   6.70  |  5.10   4.31   3.66  |  4.05   2.51   1.78  |
StDev:   0.17   0.51   1.26  |  2.07   1.05   0.99  |  1.28   0.57   0.64  |
N:         20     20     20  |    20     20     20  |    20     20     20  |


Number of parts: 3

Noise %    0%                |   15%                |   30%                |
Fpsf Len:  3      9      15  |    3      9      15  |    3      9      15  |

Avg:l)   6.00   6.00  11.00  |  5.21   5.09   4.97  |  5.35   3.94   3.30  |
StDev:   0.00   0.00   0.00  |  1.18   1.42   1.25  |  1.71   0.78   0.73  |
N:         20     20     20  |    20     20     20  |    20     20     20  |

Avg:u)   6.00   4.40   4.40  |  4.21   4.00   3.78  |  4.14   3.31   3.01  |
StDev:   0.00   0.00   0.00  |  0.89   0.89   0.84  |  0.87   0.48   0.73  |
N:         20     20     20  |    20     20     20  |    20     20     20  |

Avg:p)   6.10   5.74   6.32  |  4.65   4.78   4.91  |  4.37   3.69   3.05  |
StDev:   0.31   0.56   1.34  |  0.94   1.32   1.59  |  1.00   0.76   0.76  |
N:         20     20     20  |    20     20     20  |    20     20     20  |

Avg:r)   5.26   4.75   4.28  |  3.88   3.06   3.04  |  3.40   2.19   2.03  |
StDev:   1.69   1.08   1.03  |  1.29   0.98   1.11  |  1.00   0.79   0.87  |
N:         20     20     20  |    20     20     20  |    20     20     20  |

Avg:lp)  5.97   6.29   8.04  |  4.90   5.56   5.39  |  4.98   3.97   2.99  |
StDev:   0.11   0.49   2.08  |  0.81   1.82   1.68  |  1.49   0.74   0.63  |
N:         20     20     20  |    20     20     20  |    20     20     20  |
```

TABLE 6

```
Cache length: 128.  Part length: 11.  CHOUGH

Number of parts: 1

Noise %    0%                  |   15%                 |   30%                 |
Fpsf Len:  3      9      15    |   3      9      15    |   3      9      15    |

Avg:l)    4.40   4.40   4.40   |  3.01   1.94   1.60   |  2.07   1.20   1.04   |
StDev:    0.00   0.00   0.00   |  0.68   0.51   0.48   |  0.66   0.48   0.52   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Avg:u)    4.40   4.40   4.40   |  2.82   1.99   1.71   |  1.73   1.33   1.28   |
StDev:    0.00   0.00   0.00   |  0.76   0.57   0.44   |  0.77   0.41   0.43   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Avg:p)    4.51   4.95   4.56   |  3.01   2.03   1.72   |  2.00   1.18   1.17   |
StDev:    0.67   1.05   0.71   |  0.64   0.57   0.60   |  0.63   0.52   0.52   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Avg:r)    4.23   3.08   2.69   |  2.00   1.06   0.91   |  1.26   0.67   0.93   |
StDev:    1.90   0.81   0.76   |  0.69   0.65   0.33   |  0.67   0.53   0.56   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Avg:lp)   4.57   4.68   4.16   |  3.01   1.94   1.59   |  2.06   1.18   1.11   |
StDev:    0.39   0.75   0.80   |  0.56   0.47   0.61   |  0.65   0.58   0.55   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Number of parts: 2

Noise %    0%                  |   15%                 |   30%                 |
Fpsf Len:  3      9      15    |   3      9      15    |   3      9      15    |

Avg:l)    6.29   6.29   6.29   |  4.52   3.75   3.31   |  3.98   2.48   1.80   |
StDev:    0.00   0.00   0.00   |  1.37   0.90   0.72   |  0.95   0.48   0.48   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Avg:u)    6.29   6.29   5.50   |  3.86   3.57   3.21   |  3.54   2.44   2.03   |
StDev:    0.00   0.00   0.00   |  1.33   0.87   0.63   |  1.13   0.81   0.42   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Avg:p)    6.03   6.19   5.48   |  4.02   3.83   3.17   |  3.63   2.40   1.95   |
StDev:    0.47   1.30   0.77   |  0.85   0.95   0.48   |  0.68   0.53   0.43   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Avg:r)    4.40   3.78   4.06   |  2.38   2.29   2.31   |  2.35   1.52   1.19   |
StDev:    1.63   1.29   1.07   |  0.52   0.79   0.78   |  0.72   0.58   0.37   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Avg:lp)   6.25   6.25   6.43   |  4.60   3.95   3.34   |  3.86   2.42   1.87   |
StDev:    0.17   0.17   0.59   |  1.34   1.08   0.65   |  1.02   0.40   0.46   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Number of parts: 3

Noise %    0%                  |   15%                 |   30%                 |
Fpsf Len:  3      9      15    |   3      9      15    |   3      9      15    |

Avg:l)    6.00   6.00   6.00   |  4.81   4.37   4.53   |  4.72   3.04   3.14   |
StDev:    0.00   0.00   0.00   |  0.94   1.03   0.74   |  1.19   0.67   0.75   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Avg:u)    6.00   6.25   6.60   |  5.01   4.01   3.79   |  3.99   3.20   2.93   |
StDev:    0.00   0.00   0.00   |  1.11   0.86   0.83   |  0.83   0.68   0.54   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Avg:p)    6.10   6.14   5.96   |  4.56   4.29   4.44   |  4.37   3.57   2.92   |
StDev:    0.31   0.63   0.73   |  0.89   1.03   1.29   |  1.01   0.56   0.64   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Avg:r)    5.26   4.77   4.16   |  3.83   2.92   2.94   |  3.35   2.12   1.98   |
StDev:    1.69   1.10   0.95   |  1.37   0.90   0.92   |  1.05   0.71   0.55   |
N:          20     20     20   |    20     20     20   |    20     20     20   |

Avg:lp)   5.97   6.75   6.89   |  4.70   4.51   4.40   |  [illegible] [illegible] [illegible] |
StDev: [illegible] [illegible] 1.48 | 0.91   0.98   [illegible] | [illegible] [illegible] [illegible]
N:     [illegible] [illegible] [illegible] | 20  [illegible] [illegible] | [illegible] [illegible] [illegible]
```

TABLE 7

Cache Length: Infinite. Part length: 11. CHOUGH

Number of parts: 1

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg:l, u,p,lp)	4.40	4.40	4.40	3.05	1.93	1.68	2.04	1.27	1.19
StDev:	0.00	0.00	0.00	0.58	0.53	0.44	0.63	0.47	0.41
N:	20	20	20	20	20	20	20	20	20
Avg:r)	4.17	3.06	2.78	1.99	1.03	0.94	1.28	0.71	0.92
StDev:	1.93	0.81	0.80	0.69	0.61	0.33	0.64	0.41	0.48
N:	20	20	20	20	20	20	20	20	20

Number of parts: 2

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg:l, u,p,lp)	6.29	6.29	6.29	4.35	3.79	3.37	3.73	2.45	1.89
StDev:	0.00	0.00	0.00	1.05	1.02	0.65	0.72	0.46	0.42
N:	20	20	20	20	20	20	20	20	20
Avg:r)	4.43	3.49	4.06	2.34	2.29	2.26	2.40	1.52	1.21
StDev:	1.63	0.90	1.05	0.53	0.77	0.77	0.73	0.52	0.35
N:	20	20	20	20	20	20	20	20	20

Number of parts: 3

Noise %	0%			15%			30%		
Fpsf Len:	3	9	15	3	9	15	3	9	15
Avg:l, u,p,lp)	6.00	6.00	6.00	4.73	4.28	4.19	4.41	3.65	2.89
StDev:	0.00	0.00	0.00	1.03	0.97	0.78	0.93	0.63	0.58
N:	20	20	20	20	20	20	20	20	20
Avg:r)	5.18	4.84	4.34	3.76	2.92	2.81	3.34	2.07	1.92
StDev:	1.60	1.32	1.12	1.15	0.88	0.86	1.06	0.68	0.55
N:	20	20	20	20	20	20	20	20	20

TABLE 8

```
Cache Length: 32.  Scan: TtoB, LtoR.  Noise: 15%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6 **********
1.8 **********
2.0 ********************
2.2 **********
2.4
2.6 ********************
2.8 **********
3.0 **********
3.2
3.4
3.6
3.8
4.0

Fpsf length: 9

0.0 ******************************
0.2
0.4
0.6
0.8
1.0 **********
1.2 ********************
1.4
1.6
1.8 **********
2.0 ********************
2.2 **********
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0 **********************************************************************
0.2
0.4 **********
0.6
0.8 **********
1.0
1.2
1.4 **********
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

```
Cache Length: 64.  Scan: TtoB, Lto
                                        Noise: 15
Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6 ********************
1.8
2.0 ********************
2.2
2.4
2.6 ********************
2.8 ********************
3.0 ********************
3.2
3.4
3.6
3.8
4.0

Fpsf Length: 9

0.0 **********
0.2
0.4
0.6
0.8 **********
1.0
1.2
1.4 **********
1.6 ******************************
1.8 ********************
2.0 **********
2.2
2.4
2.6 **********
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0 ********************
0.2
0.4
0.6
0.8
1.0 **********
1.2 **********
1.4 ********************
1.6 ********************
1.8
2.0
2.2
2.4 ********************
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

Figure 4 (a), (b)

```
Cache length: 128.  Scan: TtoB, LtoR.  Noise: 15%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6 ••••••••••••••••••••
1.8
2.0 ••••••••••••••••••••
2.2 ••••••••••••••••••••
2.4
2.6 ••••••••••
2.8 ••••••••••
3.0 ••••••••••••••••••••
3.2
3.4
3.6
3.8
4.0

Fpsf length: 9

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4 ••••••••••
1.6 ••••••••••••••••••••••••••••••••••••••••••••••••••
1.8 ••••••••••••••••••••
2.0 ••••••••••
2.2 ••••••••••
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0
0.2
0.4
0.6
0.8 ••••••••••
1.0
1.2 ••••••••••••••••••••
1.4
1.6 ••••••••••••••••••••
1.8 ••••••••••••••••••••••••••••••
2.0 ••••••••••••••••••••
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

```
Cache length: Infinite.  Scan: TtoB,  LtoR.

Fpsf Length: 3
                                         Noise: 15%
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6 ••••••••••••••••••••
1.8
2.0 ••••••••••••••••••••
2.2 ••••••••••••••••••••
2.4
2.6 ••••••••••
2.8 ••••••••••
3.0 ••••••••••••••••••••
3.2
3.4
3.6
3.8
4.0

Fpsf Length: 9

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4 ••••••••••••••••••••••••••••••
1.6 ••••••••••••••••••••
1.8 ••••••••••••••••••••••••••••••••••••••••
2.0
2.2 ••••••••••
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf Length: 15

0.0
0.2
0.4
0.6
0.8
1.0
1.2 ••••••••••••••••••••••••••••••
1.4 ••••••••••
1.6 ••••••••••
1.8 ••••••••••••••••••••••••••••••
2.0 ••••••••••••••••••••
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

```
Cache Length: 32.  Scan: Random.  Noise: 15%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4 **
1.6 *******
1.8 ****
2.0 ************
2.2 ***********
2.4 ***
2.6 *******************
2.8 *********
3.0 **************
3.2
3.4 ****
3.6 ***
3.8
4.0 ***

Fpsf length: 9

0.0 ***********
0.2 *
0.4
0.6 *
0.8 *********
1.0 ******
1.2 ******
1.4 *******
1.6 *************
1.8 **********
2.0 ********
2.2 ************
2.4 ***
2.6 *****
2.8 **
3.0 *
3.2
3.4 *
3.6 ****
3.8
4.0

Fpsf length: 15

0.0 *********************************************
0.2 ****
0.4 *
0.6 **
0.8 *
1.0 **
1.2 ******
1.4 ********
1.6 *********
1.8 ******
2.0 **
2.2 *****
2.4 ****
2.6 *
2.8 *
3.0 **
3.2
3.4
3.6
3.8 *
4.0
```

```
Cache Length: 64.  Scan: Random.  Noise: 15%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6 ****************
1.8
2.0 ************
2.2 **************
2.4 *
2.6 *******************
2.8 ***********
3.0 *********************
3.2
3.4 ***
3.6 **
3.8
4.0

Fpsf length: 9

0.0
0.2
0.4
0.6
0.8 ****
1.0 ****
1.2 ******
1.4 ****************
1.6 ************
1.8 ***********************
2.0 *************
2.2 *********
2.4 ******
2.6 ****
2.8 ***
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0 ********
0.2 ***
0.4 *
0.6 ****
0.8 **
1.0 ******
1.2 **************
1.4 **********
1.6 *********************
1.8 ***********
2.0 ******
2.2 ********
2.4 *****
2.6 **
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

Figure 4 (e), (f)

```
Cache length: 128.  Scan: Random.  Noise: 15%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6 ********************
1.8
2.0 ********************
2.2 ********************
2.4
2.6 **********
2.8 **********
3.0 ********************
3.2
3.4
3.6
3.8
4.0

Fpsf length: 9

0.0
0.2
0.4
0.6
0.8
1.0 *
1.2 *****
1.4 ************
1.6 *****************************
1.8 *****************************
2.0 ********
2.2 *************
2.4 *
2.6 *
2.8 *
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0 *
0.2
0.4
0.6 *
0.8
1.0 ***
1.2 ********************
1.4 ************
1.6 *********************
1.8 ***********************
2.0 **************
2.2 *****
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

```
Cache length: Infinite.  Scan: Random.   Noise: 15%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6 ********************
1.8
2.0 ********************
2.2 ********************
2.4
2.6 **********
2.8 **********
3.0 ********************
3.2
3.4
3.6
3.8
4.0

Fpsf Length: 9

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4 ******************************
1.6 ********************
1.8 ****************************************
2.0
2.2 **********
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf Length: 15

0.0
0.2
0.4
0.6
0.8
1.0
1.2 ******************************
1.4 **********
1.6 **********
1.8 ******************************
2.0 ********************
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

```
Cache length: 128.  Scan: TtoB, LtoR.  Noise: 30%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4 ********************
1.6 ******************************
1.8 ******************************
2.0 **********
2.2
2.4 **********
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 9.

0.0
0.2
0.4
0.6
0.8 ********************
1.0 **********
1.2 ****************************************
1.4 ******************************
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0 ******************************
0.2 **********
0.4
0.6
0.8
1.0
1.2 ********************
1.4 ********************
1.6 ********************
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

```
Cache Length: Infinite.  Scan: TtoB,  LtoR.  Noise: 30%

Fpsf Length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4 ********************
1.6 ******************************
1.8 ******************************
2.0 **********
2.2
2.4 **********
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf Length: 9

0.0
0.2
0.4
0.6
0.8 **********
1.0 ********************
1.2 **************************************************
1.4 ********************
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf Length: 15

0.0
0.2
0.4
0.6
0.8 **********
1.0
1.2 ****************************************
1.4 ********************
1.6 ******************************
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

```
Cache Length: 32.  Scan: Random.  Noise: 30%

Fpsf length: 3

0.0 *******
0.2 *
0.4 *
0.6 ***
0.8 *
1.0 *******
1.2 ****
1.4 ******
1.6 **************
1.8 ****************
2.0 ********
2.2 ****
2.4 **********
2.6 ********
2.8
3.0 ******
3.2
3.4 **
3.6
3.8
4.0 **

Fpsf length: 9

0.0 *********************************************************
0.2 **
0.4 ******
0.6 ********
0.8 ***
1.0 ***
1.2 *****
1.4 ***
1.6 ****
1.8 ****
2.0 **
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0 *********************************************
0.2
0.4 **
0.6 ****
0.8 **
1.0 *********
1.2 **************
1.4 ******
1.6 *********
1.8 *****
2.0
2.2 *
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3 8
4 0
```

```
Cache Length: 64.  Scan: Random.  Noise: 30%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0 ****
1.2 *******
1.4 ***********
1.6 *****************
1.8 ********************
2.0 ***************
2.2 *****
2.4 *******
2.6 *******
2.8
3.0 ******
3.2
3.4 *
3.6
3.8
4.0

Fpsf length: 9

0.0 *****************
0.2 **
0.4 ***
0.6 ***************
0.8 *************
1.0 **********
1.2 *********************
1.4 ********
1.6 ********
1.8 ***
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0 *************************
0.2 **
0.4 ***
0.6 *
0.8 ***********
1.0 *********
1.2 ***********************
1.4 ********
1.6 **********
1.8 ******
2.0 *
2.2 *
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

```
Cache length: 128.  Scan: Random.  Noise: 30%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2 ••
1.4 •••••••••••
1.6 ••••••••••••••••••••••••••••••
1.8 ••••••••••••••••••••••••••••
2.0 •••••••••••••
2.2 •
2.4 •••••••••••
2.6 •
2.8
3.0 •
3.2
3.4
3.6
3.8
4.0

Fpsf length: 9

0.0 •••••
0.2 •
0.4 •••
0.6 •••••••••
0.8 •••••••••••••
1.0 •••••••••••••••
1.2 ••••••••••••••••••••••••••••••
1.4 •••••••••••••••
1.6 ••••••••
1.8 •
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0 ••••
0.2
0.4 ••
0.6 •••
0.8 •••••••••
1.0 •••••••••••••
1.2 •••••••••••••••••••••••••••••••
1.4 •••••••••••••••••••••
1.6 •••••••••••••
1.8 •••
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

```
Cache length: Infinite.  Scan: Random.  Noise: 30%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4 ••••••••••••••••••••
1.6 ••••••••••••••••••••••••••••••
1.8 ••••••••••••••••••••••••••••••
2.0 ••••••••••
2.2
2.4 ••••••••••
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf Length: 9

0.0
0.2
0.4
0.6
0.8 ••••••••••
1.0 ••••••••••••••••••••
1.2 ••••••••••••••••••••••••••••••••••••••••••••••••
1.4 ••••••••••••••••••••
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf Length: 15

0.0
0.2
0.4
0.6
0.8 ••••••••••
1.0
1.2 •••••••••••••••••••••••••••••••••••••••
1.4 ••••••••••••••••••••
1.6 ••••••••••••••••••••••••••••••
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

CHOUGH

```
Cache Length: 64.
Scan: TtoB, LtoR.
Noise: 15%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8 •
2.0 •
2.2 •
2.4 •
2.6 •••
2.8 •••
3.0 ••
3.2
3.4 •••
3.6 •••
3.8
4.0 •

Fpsf length: 9

0.0
0.2
0.4
0.6
0.8 •
1.0
1.2 ••
1.4 ••
1.6 •••
1.8 •••••
2.0 •
2.2 ••
2.4 •
2.6 •
2.8 •
3.0
3.2
3.4 •
3.6
3.8
4.0

Fpsf length: 15

0.0 •••••
0.2
0.4
0.6 •
0.8 •
1.0 ••
1.2 •
1.4 ••
1.6 •
1.8 •••
2.0
2.2 •••
2.4
2.6 •
2.8
3.0
3.2
3.4
3.6
3.8
4
```

```
Cache Length: 64
Scan: Random
Noise: 15%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2 •••
2.4 •
2.6 •
2.8 ••
3.0 •
3.2 •••
3.4 •••
3.6 •••
3.8
4.0 •

Fpsf length: 9

0.0
0.2
0.4
0.6 •
0.8
1.0
1.2 ••
1.4 •
1.6 ••••
1.8 ••
2.0
2.2 ••••
2.4 •
2.6 ••
2.8
3.0
3.2 ••
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0 ••
0.2
0.4
0.6 •••
0.8
1.0 •
1.2 ••
1.4 •••
1.6
1.8 ••
2.0 ••
2.2 •
2.4 ••
2.6 ••
2.8
3.0
3.2
3.4
3.6
3.8
4.0
```

```
Cache length: 64.
Scan: TtoB, LtoR.
Noise: 30%

Fpsf length: 3

0.0
0.2
0.4
0.6
0.8 •
1.0 •
1.2 ••
1.4 ••
1.6 •
1.8 ••
2.0 ••
2.2 •••
2.4 •
2.6 ••
2.8 •
3.0
3.2
3.4 •
3.6 •
3.8
4.0

Fpsf length: 9

0.0 ••••••
0.2 •
0.4 ••
0.6 •
0.8
1.0 ••
1.2 ••
1.4 ••••
1.6 •
1.8 •
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0 •••••••••••
0.2 •
0.4 •
0.6
0.8
1.0 •
1.2 •
1.4 •
1.6 ••
1.8 •
2.0 •
2.2
2.4 •
2.6
2.8
3.0
3.2
3.4
3.6
3.8
:
```

```
Cache length: 64
Scan: Random.
Noise: 30%

Fpsf length: 3

0.0
0.2
0.4
0.6 ••
0.8
1.0 •
1.2
1.4 •
1.6 ••
1.8 ••
2.0
2.2 ••••
2.4 •••
2.6 •
2.8 ••
3.0
3.2
3.4 ••
3.6
3.8
4.0

Fpsf length: 9

0.0 ••
0.2 •
0.4 ••
0.6 ••
0.8 ••
1.0 •••
1.2 •
1.4
1.6 •••
1.8 ••
2.0
2.2 •
2.4
2.6 •
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0 •••••••
0.2
0.4 •
0.6
0.8 ••
1.0 •
1.2 •••
1.4 ••••
1.6 ••
1.8
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
```

Figure 5 (a-d)

CHOUGH

```
Cache length: Infinite.
Scan: TtoB, LtoR.
Noise: 15%

Fpsf Length: 3

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8 *
2.0
2.2 **
2.4
2.6 **
2.8 **
3.0 ***
3.2 **
3.4 ***
3.6 ***
3.8 *
4.0 *

Fpsf length: 9

0.0
0.2
0.4
0.6
0.8
1.0
1.2 **
1.4 *
1.6 ****
1.8 *****
2.0 *
2.2 *****
2.4
2.6
2.8
3.0
3.2 **
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0
0.2
0.4
0.6
0.8
1.0 **
1.2 **
1.4 ****
1.6 ***
1.8 ***
2.0 **
2.2 ***
2.4
2.6 *
2.8
3.0
3.2
3.4
3.6
3.8
```

```
Cache Length: Infinite.
Scan: TtoB, LtoR.
Noise: 30%

Fpsf length 3

0.0
0.2
0.4
0.6
0.8 *
1.0 *
1.2 ***
1.4
1.6 *
1.8 *
2.0 *
2.2 ****
2.4 *
2.6 *****
2.8 *
3.0
3.2 *
3.4
3.6
3.8
4.0

Fpsf length: 9

0.0
0.2 *
0.4
0.6
0.8 ****
1.0 ****
1.2 **
1.4 **
1.6 ***
1.8 *
2.0 ***
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
4.0

Fpsf length: 15

0.0
0.2
0.4
0.6 **
0.8 ****
1.0 **
1.2 ******
1.4 **
1.6 *
1.8 **
2.0 *
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
3.8
```

Figure 5 (e), (f)

```
Cache length: 32. Scan: ItoB, LtoR.
Noise: 15%.  Instances: 1.
Parts: 1.  Fpsf length: 9.
Part length: 11.  N: 100

Flush: Threshold

Mean of Ratios:        1.212250
Std. Dev. of Ratios: 0.972063

0.0 ******************************
0.2
0.4
0.6 **
0.8 ***
1.0 *******
1.2 ************
1.4 ****
1.6 ********
1.8 *******
2.0 *******
2.2 *******
2.4 **
2.6 *****
2.8
3.0 *****
3.2
3.4
3.6 *
3.8
4.0


Flush: Random Below Threshold

Mean of Ratios:        1.578250
Std. Dev. of Ratios: 0.895189

0.0 ***************
0.2
0.4 **
0.6 ***
0.8 *
1.0 ****
1.2 ******
1.4 *******
1.6 ***********
1.8 ***********
2.0 *********
2.2 ***********
2.4 ***
2.6 **********
2.8 **
3.0 *
3.2 **
3.4 **
3.6
3.8
4.0
```

```
Cache Length: 64. Scan: ItoB, LtoR.
Noise: 15%.  Instances: 1.
Parts: 1.  Fpsf length: 9.
Part Length: 11.  N: 100

Flush: Threshold

Mean of Ratios:         1.805774
Std. Dev. of Ratios:  0.501593

0.0 *
0.2
0.4
0.6
0.8
1.0 **
1.2 ***********
1.4 ************
1.6 *************
1.8 *************************
2.0 ***********
2.2 *********
2.4 *******
2.6 ******
2.8
3.0 *
3.2 *
3.4 *
3.6
3.8
4.0


Flush: Random Below Threshold

Mean of Ratios:        1.791405
Std. Dev. of Ratios: 0.461134

0.0 *
0.2
0.4
0.6
0.8 *
1.0 ***
1.2 ********
1.4 **********
1.6 ***************
1.8 ************************
2.0 ****************
2.2 ************
2.4 ***
2.6 *****
2.8
3.0 *
3.2 *
3.4
3.6
3.8
4.0
```

Figure6 (a), (b)